

[illegible]

L 7
16-Sep-1984 02:31:54
14-Sep-1984 12:46:49

VAX-11 Bliss-32 V4.0-742
[MTAACP.SRC]REWSPC.B32;1

Page 1
(1)

```
0001 0
0002 0 MODULE REWSPC (LANGUAGE (BLISS32) ,
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1
0032 1 FACILITY: MTAACP
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module rewinds a file and spaces within a file
0037 1
0038 1 ENVIRONMENT:
0039 1
0040 1 STARLET operating system, including privileged system services
0041 1 and internal exec routines.
0042 1
0043 1 --
0044 1
0045 1
0046 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 6-AUG-1977
0047 1
0048 1 MODIFIED BY:
0049 1
0050 1 V03-003 ROW0258 Ralph O. Weber 21-NOV-1983
0051 1 The Paul Painter Memorial Enhancement
0052 1 Named for one of the unfortunate customers who suffered much
0053 1 to determine the great UCBSL_MT_RECORD secret while trying to
0054 1 create a user-written magtape driver, this change eliminates
0055 1 use of the device dependent field, UCBSL_MT_RECORD in favor of
0056 1 the device independent field, UCBSL_RECORD.
0057 1
```



```
58 0058 1 | V03-002 MMD0147 Meg Dumont, 26-Apr-1983 8:48
59 0059 1 | Change references to 80 to the symbol ANSI_LBLSZ
60 0060 1 |
61 0061 1 | V03-001 MMD0001 Meg Dumont, 3-Jan-1983 16:13
62 0062 1 | Add modifier IOSM_CLRSEREXCP to all QIO's issued by the MTAACP,
63 0063 1 | necessary for the MSCP tape drives.
64 0064 1 |
65 0065 1 | V02-004 REFORMAT Maria del C. Nasr 30-Jun-1980
66 0066 1 |
67 0067 1 | **
68 0068 1 |
69 0069 1 | LIBRARY 'SYS$LIBRARY:LIB.L32';
70 0070 1 |
71 0071 1 | REQUIRE 'SRC$:MTADEF.B32';
72 0455 1 |
73 0456 1 | FORWARD ROUTINE
74 0457 1 | REWIND_FILE : COMMON_CALL NOVALUE, ! main control for rewind file
75 0458 1 | SPACE_IN_FILE : COMMON_CALL NOVALUE, ! space within a file
76 0459 1 | SETUP_AT_END : COMMON_CALL NOVALUE, ! setup at end of file after checking position
77 0460 1 | SETUP_END : COMMON_CALL NOVALUE, ! setup at end
78 0461 1 | UPD_ST_RECORD : COMMON_CALL NOVALUE; ! update start record in current file section
79 0462 1 |
80 0463 1 | EXTERNAL ROUTINE
81 0464 1 | CLOSE_FILE : L$CLOSE_FILE, ! close file
82 0465 1 | FORMAT_FID : COMMON_CALL, ! format file ID in current VCB
83 0466 1 | GTNEXT_VOL_READ : NOVALUE JSB, ! get next volume on read
84 0467 1 | MOUNT_VOL, ! mount relative volume
85 0468 1 | POSITION_BY_FID : COMMON_CALL, ! position by FID
86 0469 1 | READ_BLOCK : COMMON_CALL, ! read data block
87 0470 1 | RESTORE_ACCESS : COMMON_CALL, ! restore original access to file
88 0471 1 | SPACE : COMMON_CALL, ! space blocks
89 0472 1 | SPACE_TM : COMMON_CALL, ! space tape marks
90 0473 1 | SYS$QIOW : ADDRESSING_MODE (ABSOLUTE);
91 0474 1 |
92 0475 1 | EXTERNAL
93 0476 1 | CURRENT_UCB : REF BBLOCK, ! address of current UCB
94 0477 1 | CURRENT_WCB : REF BBLOCK, ! address of current window control block
95 0478 1 | HDR1 : REF BBLOCK, ! address of HDR1 (EOF1) label
96 0479 1 | HDR2 : REF BBLOCK, ! address of HDR2 (EOF2) label
97 0480 1 | LOCAL_FIB : BBLOCK, ! copy of user's FIB
98 0481 1 | IO_CHANNEL,
99 0482 1 | IO_STATUS : VECTOR [2], ! IO status
100 0483 1 | USER_STATUS : VECTOR [2]; ! status returned to user
101 0484 1 |
```

```
0485 1 GLOBAL ROUTINE REWIND_FILE : COMMON_CALL NOVALUE =
0486 1
0487 1 ++
0488 1
0489 1 FUNCTIONAL DESCRIPTION:
0490 1     This routine rewinds to the beginning of the current file. if
0491 1     the beginning is on another volume, that volume is mounted and
0492 1     positioned to the beginning of the files data area. Once at the
0493 1     beginning the access to the file is reset to the original
0494 1     requested access.
0495 1
0496 1 CALLING SEQUENCE:
0497 1     REWIND_FILE()
0498 1
0499 1 INPUT PARAMETERS:
0500 1     none
0501 1
0502 1 IMPLICIT INPUTS:
0503 1     CURRENT_VCB - address of current volume control block
0504 1     CURRENT_WCB - address of current window control block
0505 1
0506 1 OUTPUT PARAMETERS:
0507 1     none
0508 1
0509 1 IMPLICIT OUTPUTS:
0510 1     none
0511 1
0512 1 ROUTINE VALUE:
0513 1     none
0514 1
0515 1 SIDE EFFECTS:
0516 1     file positioned to beginning
0517 1     original access restore
0518 1
0519 1 USER ERRORS:
0520 1     SSS_FILNOTACC - file not accessed
0521 1 --
0522 1
0523 2 BEGIN
0524 2
0525 2 EXTERNAL REGISTER
0526 2     COMMON_REG;
0527 2
0528 2 LOCAL
0529 2     FID,           ! file identification
0530 2     SEQ,           ! file section sequence number
0531 2     TM,            ! tape marks
0532 2     VOL;           ! relative volume number
0533 2
0534 2     ! if file is not accessed give error
0535 2     !
0536 2
0537 2 IF .CURRENT_WCB EQL 0
0538 2 THEN
0539 2     ERR_EXIT(SSS_FILNOTACC);
0540 2
0541 2     ! if writing, then close out file
```

```
!
IF NOT .CURRENT_WCB[WCBSV_READ]
THEN
    CLOSE_FILE();

! calculate which relative volume the beginning is on
SEQ = .CURRENT_VCB[VCBSW_CUR_SEQ];          ! file section number

IF .SEQ EQL 1
THEN
    BEGIN                                  ! currently in first file section
        IF .CURRENT_VCB[VCBSV_LOGICEOVS]
        THEN
            SPACE_TM(-4)                  ! write case
        ELSE
            BEGIN                          ! read case
                ! number of tape marks into current file section
                TM = .CURRENT_VCB[VCBSB_TM];
                IF .TM EQL 0 AND .HDR1[HD1$L_HD1LID] NEQ 'HDR1'
                THEN
                    TM = 3;
                IF .TM GEQ 1
                THEN
                    ! backspace to tape mark preceding start of data
                    SPACE_TM(-.TM);
            END;
        SPACE_TM(1);                      ! pass over TM
        HDR1[HD1$L_HD1LID] = 'HDR1';
        IF HDR2[HD2$L_HD2LID] NEQ 0
        THEN
            HDR2[HD2$L_HD2LID] = 'HDR2';
        END
    ELSE
        BEGIN
            ! current file number and section
            FID = .CURRENT_VCB[VCBSL_CUR_FID];
            FID<16, 16> = T;
            VOL = .CURRENT_VCB[VCBSB_CUR_RVN];
            VOL = .VOL - .SEQ + 1;
            POSITION_BY_FID(.FID, .VOL);
            ! want section one
            ! current volume
            ! calculate volume wanted
            ! position to file section
            IF .CURRENT_VCB[VCBSB_TM] EQL 0
```



```
217 0599 3 THEN
218 0600 SPACE_TM(1);
219 0601
220 0602
221 0603
222 0604
223 0605 1 END;
      KERNEL_CALL(RESTORE_ACCESS);
      END;
```

! restore original access to the file

```
.TITLE REWSPC
.IDENT \V04-000\
```

```
.EXTRN CLOSE_FILE, FORMAT_FID
.EXTRN GTNEXT_VOL_READ
.EXTRN MOUNT_VOL, POSITION_BY_FID
.EXTRN READ_BLOCK, RESTORE_ACCESS
.EXTRN SPACE, SPACE_TM
.EXTRN SYSSQIOW, CURRENT_UCB
.EXTRN CURRENT_WCB, HDR1
.EXTRN HDR2, LOCAL_FIB
.EXTRN IO_CHANNEL, IO_STATUS
.EXTRN USER_STATUS, SYSSCMKRNL
```

```
.PSECT $CODE$,NOWRT,2
```

```
.ENTRY REWIND_FILE, Save R2,R3,R10
```

```
MOVAB SPACE_TM, R3      : 0485
TSTL CURRENT_WCB        : 0537
BNEQ 1$                  :
CHMU #172                : 0539
MOVL CURRENT_WCB, R0     : 0544
BLBS 11(R0), 2$          :
BSBW CLOSE_FILE          : 0546
MOVZWL 38(CURRENT_VCB), SEQ : 0550
CMPL SEQ, #1             : 0552
BNEQ 7$                  :
BBC #1, 11(CURRENT_VCB), 3$ : 0556
MNEGL #4, -(SP)          : 0558
BRB 5$                   :
MOVZBL 46(CURRENT_VCB), TM : 0564
BNEQ 4$                   : 0566
CMPL @HDR1, #827475016
BEQL 4$                   :
MOVL #3, TM              : 0568
TSTL TM                  : 0570
BLEQ 6$                   :
MNEGL TM, -(SP)          : 0575
CALLS #1, SPACE_TM       :
PUSHL #1                  : 0579
CALLS #1, SPACE_TM       :
MOVL #827475016, @HDR1   : 0580
MOVL HDR2, R0            : 0582
BEQL 8$                   :
MOVL #844252232, (R0)    : 0584
BRB 8$                   : 0552
MOVL 36(CURRENT_VCB), FID : 0592
INSV #1, #16, #T6, FID   : 0593
```

51

10

52	50	2F	AB	9A	00075	MOVZBL	47(CURRENT_VCB), VOL	:	0594
	50		52	C3	00079	SUBL3	SEQ, VOL, R2	:	0595
	50	01	A2	9E	0007D	MOVAB	1(R2), VOL	:	
			50	DD	00081	PUSHL	VOL	:	0596
			51	DD	00083	PUSHL	FID	:	
	0000G	CF	02	FB	00085	CALLS	#2, POSITION_BY_FID	:	
			2E	AB	95	TSTB	46(CURRENT_VCB)-	:	0598
			05	12	0008D	BNEQ	8\$:	
			01	DD	0008F	PUSHL	#1	:	0600
	63		01	FB	00091	CALLS	#1, SPACE_TM	:	
			7E	D4	00094	CLRL	-(SP)	:	0604
			5E	DD	00096	PUSHL	SP	:	
	00000000G	9F	CF	9F	00098	PUSHAB	RESTORE_ACCESS	:	
			03	FB	0009C	CALLS	#3, @#SVSS\$CMKRNL	:	
			04	000A3		RET		:	0605

; Routine Size: 164 bytes, Routine Base: \$CODE\$ + 0000

; 224 0606 1


```
226 0607 1 ROUTINE SETUP_END (TM) : COMMON_CALL NOVALUE =
227 0608 1
228 0609 1 !++
229 0610 1
230 0611 1 FUNCTIONAL DESCRIPTION:
231 0612 1     Setup at end of file
232 0613 1
233 0614 1 CALLING SEQUENCE:
234 0615 1     SETUP_END(ARG1)
235 0616 1
236 0617 1 INPUT PARAMETERS:
237 0618 1     ARG1 - number of tape marks to be spaced and direction
238 0619 1
239 0620 1 IMPLICIT INPUTS:
240 0621 1     CURRENT_UCB - address of current unit control block
241 0622 1     HDR1 - address of 'HDR1' and 'EOF1' label
242 0623 1
243 0624 1 OUTPUT PARAMETERS:
244 0625 1     none
245 0626 1
246 0627 1 IMPLICIT OUTPUTS:
247 0628 1     CURRENT_VCB[VCB$L_ST_RECORD]
248 0629 1
249 0630 1 ROUTINE VALUE:
250 0631 1     none
251 0632 1
252 0633 1 SIDE EFFECTS:
253 0634 1     none
254 0635 1
255 0636 1 !--
256 0637 1
257 0638 2 BEGIN
258 0639 2
259 0640 2 EXTERNAL REGISTER
260 0641 2     COMMON_REG;
261 0642 2
262 0643 2 EXTERNAL ROUTINE
263 0644 2     LIB$CVT_DTB      : ADDRESSING_MODE (ABSOLUTE);
264 0645 2
265 0646 2 LOCAL
266 0647 2     BLOCK;
267 0648 2
268 0649 2 SPACE_TM(.TM);      ! space to end of file, right before end date TM
269 0650 2
270 0651 2 ! setup as if trailers had not been read
271 0652 2
272 0653 2 HDR1[HD1$L_HD1LID] = 'HDR1';
273 0654 2
274 0655 2 IF HDR2[HD2$L_HD2LID] NEQ 0
275 0656 2 THEN
276 0657 2     HDR2[HD2$L_HD2LID] = 'HDR2';
277 0658 2
278 0659 2 IF NOT LIB$CVT_DTB(E01$$_BLOCKCNT, HDR1[E01$T_BLOCKCNT], BLOCK)
279 0660 2 THEN
280 0661 2     ERR_EXIT(SS$_BLOCKCNTERR);
281 0662 2
282 0663 2 BLOCK = .CURRENT_UCB[UCB$L_RECORD] - .BLOCK;
```

```
: 283
: 284      0664 2  KERNEL_CALL(UPD_ST_RECORD, .BLOCK);
      0665 1  END;
```

```
                                .EXTRN LIB$CVT_DTB
                                0000 00000 SETUP_END:
                                .WORD  Save nothing
                                SUBL2  #4, SP
                                PUSHL  TM
                                CALLS  #1, SPACE TM
                                MOVL   #827475018, @HDR1
                                MOVL   HDR2, R0
                                BEQL   1$
                                MOVL   #844252232, (R0)
                                PUSHL  SP
                                ADDL3  #54, HDR1, -(SP)
                                PUSHL  #6
                                CALLS  #3, @LIB$CVT_DTB
                                BLBS   R0, 2$
                                CHMU   #2368
                                MOVL   CURRENT_UCB, R0
                                SUBL3  BLOCK, T76(R0), BLOCK
                                PUSHL  BLOCK
                                PUSHL  #1
                                PUSHL  SP
                                PUSHAB UPD_ST_RECORD
                                CALLS  #4, @SYS$CMKRNL
                                RET
                                : 0607
                                : 0649
                                : 0653
                                : 0655
                                : 0657
                                : 0659
                                :
                                :
                                : 0661
                                : 0663
                                : 0664
                                :
                                : 0665
```

; Routine Size: 89 bytes, Routine Base: \$CODE\$ + 00A4

```

0666 1 ROUTINE UPD_ST_RECORD (BLOCK) : COMMON_CALL NOVALUE =
0667 1
0668 1 ++
0669 1
0670 1 FUNCTIONAL DESCRIPTION:
0671 1 This routine updates the start record count in the volume control block
0672 1 and sets the TM count to 1 because now positioned before end data TM
0673 1
0674 1 CALLING SEQUENCE:
0675 1 UPD_ST_RECORD(ARG1)
0676 1 called in kernel mode
0677 1
0678 1 INPUT PARAMETERS:
0679 1 ARG1 - new value of start record count
0680 1
0681 1 IMPLICIT INPUTS:
0682 1 CURRENT_VCB
0683 1
0684 1 OUTPUT PARAMETERS:
0685 1 none
0686 1
0687 1 IMPLICIT OUTPUTS:
0688 1 CURRENT_VCB[VCB$$_ST_RECORD] = BLOCK
0689 1
0690 1 ROUTINE VALUE:
0691 1 none
0692 1
0693 1 SIDE EFFECTS:
0694 1 none
0695 1
0696 1 --
0697 1
0698 1 BEGIN
0699 1
0700 1 EXTERNAL REGISTER
0701 1 COMMON_REG;
0702 1
0703 1 CURRENT_VCB[VCB$$_TM] = 1;
0704 1 CURRENT_VCB[VCB$$_ST_RECORD] = .BLOCK;
0705 1 END;

```

```

0000 00000 UPD_ST_RECORD:
2E AB 01 90 00002 .WORD Save nothing
30 AB 04 AC D0 00006 MOVBL #1, 46(CURRENT_VCB)
04 0000B RET BLOCK, 48(CURRENT_VCB)

```

; Routine Size: 12 bytes, Routine Base: \$CODE\$ + 00FD

; 326 0706 1

```

: 0666
: 0703
: 0704
: 0705

```



```
0707 1 GLOBAL ROUTINE SPACE_IN_FILE : COMMON_CALL NOVALUE =
0708 1
0709 1 ++
0710 1
0711 1 FUNCTIONAL DESCRIPTION:
0712 1     This routine spaces forwards and backwards within a file
0713 1
0714 1 CALLING SEQUENCE:
0715 1     SPACE_IN_FILE()
0716 1
0717 1 INPUT PARAMETERS:
0718 1     none
0719 1
0720 1 IMPLICIT INPUTS:
0721 1     CURRENT_VCB - address of current volume control block
0722 1     LOCAL_FIB - copy of user's file information block
0723 1
0724 1 OUTPUT PARAMETERS:
0725 1     none
0726 1
0727 1 IMPLICIT OUTPUTS:
0728 1     none
0729 1
0730 1 ROUTINE VALUE:
0731 1     none
0732 1
0733 1 SIDE EFFECTS:
0734 1     none
0735 1
0736 1 USER ERRORS:
0737 1     $$$_BEGOFFILE - beginning of file
0738 1     $$$_ENDOFFILE - end of file
0739 1     $$$_FILNOTACC - file not accessed
0740 1     $$$_BADPARAM - can not space forward if writing
0741 1     $$$_TAPEPOSLOST - tape position lost
0742 1 --
0743 1
0744 2 BEGIN
0745 2
0746 2 EXTERNAL REGISTER
0747 2     COMMON_REG;
0748 2
0749 2 STACKLOCAL
0750 2     BLOCKS;                ! number of blocks to space
0751 2
0752 2 LOCAL
0753 2     TM;                    ! number of tape marks into file section
0754 2
0755 2     ! file must be accessed
0756 2     !
0757 2
0758 2 IF .CURRENT_WCB EQL 0
0759 2 THEN
0760 2     ERR_EXIT(SS$_FILNOTACC);
0761 2
0762 2 BLOCKS = .LOCAL_FIB[FIB$_CNTRLVAL];
0763 2
```

```
385 0764
386 0765
387 0766
388 0767
389 0768
390 0769
391 0770
392 0771
393 0772
394 0773
395 0774
396 0775
397 0776
398 0777
399 0778
400 0779
401 0780
402 0781
403 0782
404 0783
405 0784
406 0785
407 0786
408 0787
409 0788
410 0789
411 0790
412 0791
413 0792
414 0793
415 0794
416 0795
417 0796
418 0797
419 0798
420 0799
421 0800
422 0801
423 0802
424 0803
425 0804
426 0805
427 0806
428 0807
429 0808
430 0809
431 0810
432 0811
433 0812
434 0813
435 0814
436 0815
437 0816
438 0817
439 0818
440 0819
441 0820

IF .BLOCKS GTR 0
THEN
    BEGIN
        ! sign determines direction to space
        ! beginning of forward space
        IF .BLOCKS<16, 16> NEQ 0
        THEN
            ERR_EXIT(SS$_BADPARAM);
        ! can not space forward if writing
        !
        IF NOT .CURRENT_WCB[WCB$_V_READ]
        THEN
            ERR_EXIT(SS$_BADPARAM);
        ! position to data in current file section
        !
        IF .CURRENT_VCB[VCB$_B_TM] EQL 0
        AND
        .HDR1[HD1$_L_HD1LID] EQL 'HDR1'
        THEN
            SPACE_TM(1);
        IF .CURRENT_VCB[VCB$_B_TM] NEQ 1
        THEN
            BEGIN
                IF .CURRENT_VCB[VCB$_B_TM] EQL 2
                THEN
                    TM = -1
                ELSE
                    BEGIN
                        IF .CURRENT_VCB[VCB$_V_LOGICEOVS]
                        THEN
                            TM = -3
                        ELSE
                            TM = -2;
                    END;
                SPACE_TM(.TM);
            END;
        WHILE 1
        DO
            BEGIN
                ! forward space loop
                IF SPACE(.BLOCKS)
                THEN
                    EXITLOOP;
                USER STATUS<16, 16> = .USER STATUS<16, 16> + .IO STATUS<16, 16> - 1;
                BLOCKS = .BLOCKS - .IO STATUS<16, 16> + 1; ! TM counts
                IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
                THEN
```

```
442      ERR_EXIT(SS$_TAPEPOSLOST);
443
444      IF .HDR1[HD1$L_HD1LID] EQL 'EOF1'
445      THEN
446          BEGIN
447              SETUP_END(-1);
448              KERNEL CALL(RESTORE_ACCESS);
449              ERR_EXIT(SS$_ENDOFFILE);
450              END;
451
452      IF .HDR1[HD1$L_HD1LID] NEQ 'EOV1'
453      THEN
454          ERR_EXIT(SS$_TAPEPOSLOST);
455
456      GTNEXT_VOL_READ();          ! get next volume in volume set
457
458      IF .CURRENT_VCB[VCB$B_TM] EQL 0
459      THEN
460          SPACE_TM(1);
461
462      END;
463
464      ELSE                          ! end of forward space loop
465      BEGIN                          ! begin of backspace
466
467          IF .BLOCKS NEQ 0
468          THEN
469              BLOCKS = -(.BLOCKS);
470
471          IF .BLOCKS<15, 17> NEQ 0
472          THEN
473              ERR_EXIT(SS$_BADPARAM);
474
475          ! position to data if not there
476          !
477
478          IF NOT .CURRENT_WCB[WCB$V_READ]
479          THEN
480              CLOSE_FILE();
481
482          IF .CURRENT_VCB[VCB$V_LOGICEOVS]
483          THEN
484              SETUP_END(-3)
485          ELSE
486              BEGIN                      ! read case
487                  TM = .CURRENT_VCB[VCB$B_TM];
488
489                  IF .TM EQL 0 AND .HDR1[HD1$L_HD1LID] EQL 'HDR1'
490                  THEN
491                      BEGIN
492                          SPACE_TM(1);
493                          KERNEL CALL(RESTORE_ACCESS);
494                          ERR_EXIT(SS$_BEGOFFILE);
495                          END;
496
497                  IF .TM EQL 0
```



```
      THEN
        TM = 3;
      IF .TM GEQ 2
      THEN
        SETUP_END(-(TM - 1));
      END;
    WHILE 1
    DO
      BEGIN
        ! beginning of backspace loop
        LOCAL
          FID,
          SEQ,
          VOL;
        IF SPACE(-(BLOCKS))
        THEN
          EXITLOOP;
        USER_STATUS<16, 16> = .USER_STATUS<16, 16> + .IO_STATUS<16, 16> - 1;
        ! calc number remaining to space
        !
        BLOCKS = .BLOCKS - .IO STATUS<16, 16> + 1;
        FID = .CURRENT_VCB[VCB$$_CUR_FID];
        SEQ = .CURRENT_VCB[VCB$$_CUR_SEQ];
        IF .SEQ EQL 1
        THEN
          ! is tape positioned at beginning of file?
          BEGIN
            ! space to beginning of data to avoid blocking virtual IO
            !
            SPACE TM(1);
            KERNEL_CALL(RESTORE_ACCESS);
            ERR_EXIT(SS$_BEGOFFILE);
            END;
          ! previous volume number
          !
          VOL = .CURRENT_VCB[VCB$$_CUR_RVN] - 1;
          MOUNT_VOL(.VOL, $FIELDMASK(MOUSV_LBLCHECK));
          BEGIN
            LOCAL
              STATUS;
            STATUS = SYS$QIOW(EFN, .IO_CHANNEL, IOS_SENSECHAR OR IOSM_CLSEREXCP,
              IO_STATUS, 0,0,0,0,0,0,0,0);
            IF .STATUS AND .IO_STATUS EQL SS$_ENDOFTAPE
            THEN
              BEGIN
```

```

556 0935 6 SPACE_TM(-3);
557 0936 6 SPACE_TM(1);
558 0937 6 SETUP_AT_END();
559 0938 6 KERNEL_CALL(FORMAT_FID, CURRENT_VCB[VCBSL_CUR_FID]);
560 0939 6 END
561 0940 6 ELSE
562 0941 6 BEGIN
563 0942 6 FID<16, 16> = .SEQ - 1;
564 0943 6 POSITION_BY_FID(.FID, .VOL);
565 0944 6 TM = 2 - .CURRENT_VCB[VCBSB_TM];
566 0945 6 SPACE_TM(TM);
567 0946 6 SETUP_AT_END();
568 0947 6 END;
569 0948 6
570 0949 6 END;
571 0950 6 ! end of while loop
572 0951 6
573 0952 6 END;
574 0953 6 ! end of forward and backward space
575 0954 6
576 0955 6 USER_STATUS<16, 16> = .USER_STATUS<16, 16> + .IO_STATUS<16, 16>;
577 0956 6 KERNEL_CALL(RESTORE_ACCESS);
577 0956 6 END;

```

			07FC 00000	.ENTRY	SPACE_IN_FILE, Save R2,R3,R4,R5,R6,R7,R8,-	0707
5A	0000G	CF	9E 00002	MOVAB	R9,R10	
59	0000G	CF	9E 00007	MOVAB	RESTORE_ACCESS, R10	
58	0000G	CF	9E 0000C	MOVAB	HDR1, R9	
57	0000G	CF	9E 00011	MOVAB	IO STATUS, R8	
56	00000000G	9F	9E 0C016	MOVAB	SPACE_TM, R7	
5E		04	C2 0001D	MOVAB	@#SYSSCMKRN, R6	
	0000G	CF	D5 00020	SUBL2	#4, SP	
		04	12 00024	TSTL	CURRENT_WCB	0758
	00AC	8F	BF 00026	BNEQ	1\$	
6E	0000G	CF	D0 0002A	CHMU	#172	0760
50		6E	D0 0002F	MOVL	LOCAL_FIB+24, BLOCKS	0762
		03	14 00032	MOVL	BLOCKS, R0	0764
		00BF	31 00034	BGTR	2\$	
	02	AE	B5 00037	BRW	15\$	
		02	13 0003A	TSTW	BLOCKS+2	0768
		14	BF 0003C	BEQL	3\$	
50	0000G	CF	D0 0003E	CHMU	#20	0770
02	0B	A0	E8 00043	MOVL	CURRENT_WCB, R0	0775
		14	BF 00047	BLBS	11(R0), -4\$	
	2E	AB	95 00049	CHMU	#20	0777
		0F	12 0004C	TSTB	46(CURRENT_VCB)	0782
31524448	8F	00	B9 D1 0004E	BNEQ	5\$	
		05	12 00056	CMP	@HDR1, #827475016	0784
		01	DD 00058	BNEQ	5\$	
67		01	FB 0005A	PUSHL	#1	0786
01	2E	AB	91 0005D	CALLS	#1, SPACE_TM	
		1D	13 00061	CMPB	46(CURRENT_VCB), #1	0788
02	2E	AB	91 00063	BEQL	10\$	
				CMPB	46(CURRENT_VCB), #2	0792

		52		05	12	00067	BNEQ	6\$			
		01	CE	00069			MNEGL	#1, TM		0794	
		00	11	0006C			BRB	8\$			
	05	08	AB	01	E1	0006E	6\$:	BBC	#1, 11(CURRENT_VCB), 7\$	0798	
		52		03	CE	00073		MNEGL	#3, TM	0800	
				03	11	00076		BRB	8\$		
		52		02	CE	00078	7\$:	MNEGL	#2, TM	0802	
				52	DD	0007B	8\$:	PUSHL	TM	0805	
		67		01	FB	0007D	9\$:	CALLS	#1, SPACE_TM		
				6E	DD	00080	10\$:	PUSHL	BLOCKS	0812	
		0000G	CF	01	FB	00082		CALLS	#1, SPACE		
		03		50	E9	00087		BLBC	RO, 11\$		
				018E	31	0008A		BRW	28\$		
		50		0000G	CF	3C	0008D	11\$:	MOVZWL	USER_STATUS+2, RO	0816
		51		02	AB	3C	00092		MOVZWL	IO_STATUS+2, R1	
		50			51	CO	00096		ADDL2	R1, RO	
0000G	CF	50			01	A3	00099		SUBW3	#1, RO, USER_STATUS+2	
		50		02	AB	3C	0009F		MOVZWL	IO_STATUS+2, RO	0817
	50	6E			50	C3	000A3		SUBL3	RO, BLOCKS, RO	
		6E		01	A0	9E	000A7		MOVAB	1(RO), BLOCKS	
		7E		50	8F	9A	000AB		MOVZBL	#80, -(SP)	0819
					69	DD	000AF		PUSHL	HDR1	
		0000G	CF		02	FB	000B1		CALLS	#2, READ_BLOCK	
		04			50	E8	000B6		BLBS	RO, 12\$	
				0224	8F	BF	000B9		CHMU	#548	0821
	31464F45	8F		00	89	D1	000BD	12\$:	CMPL	@HDR1, #826691397	0823
					15	12	000C5		BNEQ	13\$	
		7E			01	CE	000C7		MNEGL	#1, -(SP)	0826
		FECC	CF		01	FB	000CA		CALLS	#1, SETUP_END	
					7E	D4	000CF		CLRL	-(SP)	0827
				4400	8F	BB	000D1		PUSHR	#*M<R10, SP>	
					03	FB	000D5		CALLS	#3, SYS\$CMKRN	
		66			8F	BF	000D8		CHMU	#2160	0828
	31564F45	8F		0870	89	D1	000DC	13\$:	CMPL	@HDR1, #827739973	0831
				00	04	13	000E4		BEQL	14\$	
				0224	8F	BF	000E6		CHMU	#548	0833
					0000G	30	000EA	14\$:	BSBW	GTNEXT VOL_READ	0835
				2E	AB	95	000ED		TSTB	46(CURRENT_VCB)	0837
					8E	12	000F0		BNEQ	10\$	
					01	DD	000F2		PUSHL	#1	0839
					87	11	000F4		BRB	9\$	
					03	13	000F6	15\$:	BEQL	16\$	0847
		6E			50	CE	000F8		MNEGL	RO, BLOCKS	0849
00	01	AE			07	ED	000FB	16\$:	CMPZV	#7, #17, BLOCKS+1, #0	0851
					02	13	00101		BEQL	17\$	
					14	BF	00103		CHMU	#20	0853
		50		0000G	CF	D0	00105	17\$:	MOVL	CURRENT_VCB, RO	0858
		03		0B	A0	E8	0010A		BLBS	11(RO), 18\$	
					0000G	30	0010E		BSBW	CLOSE FILE	0860
					01	E1	00111	18\$:	BBC	#1, 1T(CURRENT_VCB), 19\$	0862
		05	08		03	CE	00116		MNEGL	#3, -(SP)	0864
		7E			34	11	00119		BRB	22\$	
					AB	9A	0011B	19\$:	MOVZBL	46(CURRENT_VCB), TM	0867
		52		2E	1C	12	0011F		BNEQ	20\$	0869
					8F	D1	00121		CMPL	@HDR1, #827475016	
	31524448			00	12	12	00129		BNEQ	20\$	
					01	DD	0012B		PUSHL	#1	0872

67		01	FB	0012D	CALLS	#1, SPACE_TM	
		7E	D4	00130	CLRL	-(SP)	0873
	4400	8F	BB	00132	PUSHR	#^M<R10, SP>	
66		03	FB	00136	CALLS	#3, SYS\$CMKRNL	
	0938	8F	BF	00139	CHMU	#2360	0874
		52	D5	0013D	TSTL	TM	0877
		03	12	0013F	BNEQ	21\$	
52		03	D0	00141	MOVL	#3, TM	0879
02		52	D1	00144	CMPL	TM, #2	0881
		0B	19	00147	BLSS	23\$	
	FF	A2	9F	00149	PUSHAB	-1(TM)	0883
6E		6E	CE	0014C	MNEGL	(SP), (SP)	
FE47		01	FB	0014F	CALLS	#1, SETUP_END	
		6E	CE	00154	MNEGL	BLOCKS, -7(SP)	0896
0000G		01	FB	00157	CALLS	#1, SPACE	
		50	E9	0015C	BLBC	R0, 24\$	
		00B9	31	0015F	BRW	28\$	
50	0000G	CF	3C	00162	MOVZWL	USER_STATUS+2, R0	0900
51	02	AB	3C	00167	MOVZWL	IO_STATUS+2, R1	
50		51	C0	0016B	ADDL2	R1, R0	
50		01	A3	0016E	SUBW3	#1, R0, USER_STATUS+2	
50	02	AB	3C	00174	MOVZWL	IO_STATUS+2, R0	0904
6E		50	C3	00178	SUBL3	R0, BLOCKS, R0	
6E	01	A0	9E	0017C	MOVAB	1(R0), BLOCKS	
55	24	AB	D0	00180	MOVL	36(CURRENT_VCB), FID	0905
54	26	AB	3C	00184	MOVZWL	38(CURRENT_VCB), SEQ	0906
01		54	D1	00188	CMPL	SEQ, #1	0908
		12	12	0018B	BNEQ	25\$	
		01	DD	0018D	PUSHL	#1	0915
67		01	FB	0018F	CALLS	#1, SPACE_TM	
		7E	D4	00192	CLRL	-(SP)	0916
	4400	8F	BB	00194	PUSHR	#^M<R10, SP>	
66		03	FB	00198	CALLS	#3, SYS\$CMKRNL	
	0938	8F	BF	0019B	CHMU	#2360	0917
53		AB	9A	0019F	MOVZBL	47(CURRENT_VCB), VOL	0922
	2F	02	DD	001A3	PUSHL	#2	0923
		73	9F	001A5	PUSHAB	-(VOL)	
0000G	CF	02	FB	001A7	CALLS	#2, MOUNT_VOL	
		7E	7C	001AC	CLRQ	-(SP)	0929
		7E	7C	001AE	CLRQ	-(SP)	
		7E	7C	001B0	CLRQ	-(SP)	
		7E	7C	001B2	CLRQ	-(SP)	
		58	DD	001B4	PUSHL	R8	
7E	021B	8F	3C	001B6	MOVZWL	#539, -(SP)	
	0000G	CF	DD	001BB	PUSHL	IO_CHANNEL	
		01	DD	001BF	PUSHL	#1	
00000000G	9F	0C	FB	001C1	CALLS	#12, @#SYSSQIOW	
	29	50	E9	001C8	BLBC	STATUS, 26\$	0932
00000878	8F	68	D1	001CB	CMPL	IO_STATUS, #2168	
		20	12	001D2	BNEQ	26\$	
		03	CE	001D4	MNEGL	#3, -(SP)	0935
7E		01	FB	001D7	CALLS	#1, SPACE_TM	
67		01	DD	001DA	PUSHL	#1	0936
		01	FB	001DC	CALLS	#1, SPACE_TM	
0000V	CF	00	FB	001DF	CALLS	#0, SETUP-AT_END	0937
		AB	9F	001E4	PUSHAB	36(CURRENT_VCB)	0938
	24	01	DD	001E7	PUSHL	#1	

55	10	66	0000G	5E DD 001E9	PUSHL SP	:	
				CF 9F 001EB	PUSHAB	:	
				04 FB 001EF	CALLS #4, SYSSCMKRNL	:	
		50	FF	24 11 001F2	BRB 27\$:	0932
		10		A4 9E 001F4	MOVAB -1(R4), R0	:	0942
				50 FO 001F8	INSV R0, #16, #16, FID	:	
				53 DD 001FD	PUSHL VOL	:	0943
				55 DD 001FF	PUSHL FID	:	
	0000G	CF		02 FB 00201	CALLS #2, POSITION_BY_FID	:	
		52	2E	AB 9A 00206	MOVZBL 46(CURRENT_VCB), TM	:	0944
	52	02		52 C3 0020A	SUBL3 TM, #2, TM	:	
				52 DD 0020E	PUSHL TM	:	0945
		67		01 FB 00210	CALLS #1, SPACE_TM	:	
	0000V	CF		00 FB 00213	CALLS #0, SETUP_AT_END	:	0946
				FF 39 31 00218	BRW 23\$:	0887
	0000G	CF	02	A8 A0 0021B	ADDW2 IO STATUS+2, USER_STATUS+2	:	0954
				7E D4 00221	CLRL -(SP)	:	0955
			4400	8F BB 00223	PUSHR #^M<R10, SP>	:	
		66		03 FB 00227	CALLS #3, SYSSCMKRNL	:	
				04 0022A	RET	:	0956

; Routine Size: 555 bytes, Routine Base: \$CODE\$ + 0109

; 578 0957 1

```
580 0958 1 ROUTINE SETUP_AT_END : COMMON_CALL NOVALUE =
581 0959 1
582 0960 1 ++
583 0961 1
584 0962 1 FUNCTIONAL DESCRIPTION:
585 0963 1 This routine makes the current file section current
586 0964 1 and positions at end of this file section's data
587 0965 1
588 0966 1 CALLING SEQUENCE:
589 0967 1 SETUP_AT_END()
590 0968 1
591 0969 1 INPUT PARAMETERS:
592 0970 1 none
593 0971 1
594 0972 1 IMPLICIT INPUTS:
595 0973 1 none
596 0974 1
597 0975 1 OUTPUT PARAMETERS:
598 0976 1 none
599 0977 1
600 0978 1 IMPLICIT OUTPUTS:
601 0979 1 file section made current
602 0980 1 start record of data section calculated
603 0981 1
604 0982 1 ROUTINE VALUE:
605 0983 1 none
606 0984 1
607 0985 1 SIDE EFFECTS:
608 0986 1 none
609 0987 1
610 0988 1 --
611 0989 1
612 0990 2 BEGIN
613 0991 2
614 0992 2 EXTERNAL REGISTER
615 0993 2 COMMON_REG;
616 0994 2
617 0995 2 IF NOT READ_BLOCK(.HDR1, ANSI_LBLSZ)
618 0996 2 THEN
619 0997 2 ERR_EXIT(SS$_TAPEPOSLOST);
620 0998 2
621 0999 2 IF .HDR1[E01$L_E01LID] NEQ 'EOV1'
622 1000 2 THEN
623 1001 2 ERR_EXIT(SS$_TAPEPOSLOST);
624 1002 2
625 1003 2 SETUP_END(-1);
626 1004 1 END;
```

```
0000 00000 SETUP_AT_END:
7E 50 8F 9A 00002 .WORD Save nothing
0000G CF DD 00006 MOVZBL #80, -(SP)
0000G CF 02 FB 0000A PUSHL HDR1
CALLS #2, READ_BLOCK
```

```
0958
0995
...
```



```

      04      50 E8 0000F      BLBS      R0, 1$
31564F45 8F      0224 8F BF 00012      CHMU      #548
      0000G DF D1 00016 1$:      CMPL      @HDR1, #827739973
      0224 04 13 0001F      BEQL      2$
      8F BF 00021      CHMU      #548
      7E      01 CE 00025 2$:      MNEGL     #1, -(SP)
      FD43 CF      01 FB 00028      CALLS    #1, SETUP_END
      04 0002D      RET
  
```

```

: 0997
: 0999
: 1001
: 1003
: 1004
  
```

; Routine Size: 46 bytes, Routine Base: \$CODE\$ + 0334

```

: 627      1005 1 END
: 628      1006 1
: 629      1007 0 ELUDOM
  
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	866	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	Symbols		Pages Mapped	Processing Time
	Total	Loaded		
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	28	1000	00:01.9

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:REWSPC/OBJ=OBJ\$:REWSPC MSRC\$:REWSPC/UPDATE=(ENH\$:REWSPC)

```

: Size:      866 code + 0 data bytes
: Run Time:   00:17.9
: Elapsed Time: 00:37.9
: Lines/CPU Min: 3367
: Lexemes/CPU-Min: 17073
: Memory Used: 197 pages
: Compilation Complete
  
```


0256

AH-BT13A-SE
VAX/VMS V4.0

**DIGITAL
CONFIDE**

EQUIPMENT
NTIAL AND

CORPORATION
PROPRIETARY